IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Kenneth Charles Boydstun, et al.	§	
		§	Group Art Unit: 3685
Serial No.:	10/075,336	§	
		§	Examiner: Sherr, Cristina O.
Filed:	February 13, 2002	§	
		§	Confirmation No.: 8714
For:	METHOD AND SOFTWARE FOR MIGRATING PROTECTED AUTHENTICATION DATA	§	

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sirs:

This Appeal Brief is filed in support for the appeal in the above referenced application and is filed pursuant to the Notice of Appeal filed June 21, 2010 and the Notice of Panel Decision dated August 3, 2010 issued in response to Appellants' Pre-Appeal Brief Request for Review and Reasons for Request filed June 21, 2010. Appellants authorize all required fees under 37 C.F.R. § 1.17 to be charged to Deposit Account No. 21-0765, of Sprint Communications Company L.P.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest in the present application is the following party: Sprint Communications Company L.P.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

A. Total Number of Claims in the Application

The claims in the application are: 22-35

B. Status of All Claims in the Application

1. Claims canceled: 1-15 and 19-21
2. Claims withdrawn from consideration but not canceled: 16-18
3. Claims pending: 22-35
4. Claims allowed: None
5. Claims rejected: 22-35

C. Claims on Appeal

The claims on appeal are: 22-35

IV. STATUS OF AMENDMENTS

No amendments were filed after the March 31, 2010 Final Office Action (hereinafter "Final Office Action").

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present disclosure provides a method and software for migrating from a first authentication and authorization system to a second while minimizing the impact on end users and applications. *See, e.g.,* Application at 7, ¶ [0016], lines 1-3.[1] One of the most difficult aspects of migration of an authentication system is migration of the authentication information for the individual users (whether the users are persons or applications or other computers or networks). *See, e.g.,* Application at 8, ¶ [0018], lines 5-7. While one could always simply change out the authentication system and have every user re-register to provide new security information, this involves substantial coordination including safeguards that the authorized user is the one providing the new information. *See, e.g.,* Application at 8, ¶ [0018], lines 7-10. It also causes significant additional effort by the end users, while a more transparent migration reduces end user frustration. *See, e.g.,* Application at 8, ¶ [0018], lines 10-11. Finally, in a simple world one could replicate or transform the security datastore to a datastore for the new system, porting the information across all at once and having it available for the new authenticator to use. *See, e.g.,* Application at 8, ¶ [0018], lines 11-13. However, the custom nature of some of the schema and the various encryption efforts make this task highly challenging to impossible for some migrations. *See, e.g.,* Application at 8, ¶ [0018], lines 13-15.

In one embodiment, the first step is to migrate the data that is not encrypted first. *See, e.g.,* Application at 9, ¶ [0021], lines 3-4. This is done in a traditional replication or transformation

[1] 37 C.F.R. § 41.37 (c)(1)(v) provides that the "[s]ummary of claimed subject matter . . . shall refer to the specification by page and line number." The instant application was presented in numbered page and numbered paragraph form. As such, the citations to the specification support for the claimed subject matter will be presented in the following form: Application at ___ (page number), ¶ [____] (paragraph number), lines ____ (lines within the corresponding paragraph). On the occasion when the pertinent paragraph is contained on multiple pages, the paragraph line numbering will begin anew on subsequent pages.

between corresponding fields in the two datastores, (the source datastore and the target datastore). *See, e.g.,* Application at 9, ¶ [0021], lines 4-5. The main field that is of concern that is encrypted is the password. *See, e.g.,* Application at 9, ¶ [0021], lines 5-6. Since it is stored encrypted, it cannot simply be copied into an LDAP datastore and be expected to work. *See, e.g.,* Application at 9, ¶ [0021], lines 6-7.

Thus, after the unencrypted data is converted and stored in the target datastore, an identification and password is received from a user seeking access to information protected by the target user authenticator. *See, e.g.,* Application at 4, ¶ [0008], lines 1-2, and at 5, ¶ [0008], lines 2-3. The corresponding identification in the target datastore is located and it is determined whether the target datastore includes a password associated with the identification. *See, e.g.,* Application at 4, ¶ [0008], line 2 to 5, ¶ [0008], line 2. If the target datastore does not include a password associated with the identification, then the received identification and received password are submitted to the source user authenticator. *See, e.g.,* Application at 5, ¶ [0008], lines 3-5. The source user authenticator is then monitored for an approval response. *See, e.g.,* Application at 5, ¶ [0008], line 5. On receipt of an approval response from the source user authenticator, the target datastore is populated with the received password associating the received password with the corresponding identification. *See, e.g.,* Application at 5, ¶ [0008], lines 5-7. Finally, the identification and password are authenticated using the target user authenticator. *See, e.g.,* Application at 5, ¶ [0008], lines 7-8.

Claim 22 is independent and is directed to a method for populating password data to a target datastore of a target user authenticator after migration from a source user authenticator having a source datastore, while also responding to user requests for information, *see, e.g.,* Application at 9, ¶ [0022], line 1 to 10, ¶ [0022], line 6 and at 12, ¶ [0029], lines 1-7, the method

comprising: migrating the source datastore to the target datastore, *see, e.g.,* Application at 11, ¶ [0028], lines 1-2, wherein the source datastore comprises user identification data and user authentication data, wherein the source datastore is associated with a source user authenticator, *see, e.g.,* Application at 7, ¶ [0016], lines 5-8, wherein the target datastore is associated with a target user authenticator, *see, e.g.,* Application at 8, ¶ [0018], lines 4-5, and wherein the target user authenticator is in communication with the source user authenticator, *see, e.g.,* Application at 11, ¶ [0028], lines 1-3; intercepting, with an interceptor, a request to the source user authenticator from a user seeking access to information protected by the target user authenticator, wherein the interceptor prompts the user for an identification, *see, e.g.,* Application at 12, ¶ [0031], lines 1-3, receives the identification from the user, locates a corresponding identification in the target datastore, and searches the target datastore for a user authentication data associated with the corresponding identification, *see, e.g.,* Application at 12, ¶ [0031], lines 3-6*;* forwarding the original intercepted request to the source user authenticator upon determining that the target datastore does not include a user authentication data associated with the corresponding identification*, see, e.g.,* Application at 13, ¶ [0033], lines 1-4; using the source user authenticator to prompt for and receive the identification and user authentication data from the user, *see, e.g.,* Application at 13, ¶ [0033], lines 4-9, the target user authenticator: monitors the source user authenticator for an approval response, *see, e.g.,* Application at 13, ¶ [0035], lines 1-3, and upon an approval response from the source user authenticator, captures the user authentication data provided to the source user authenticator by the user, populates the target datastore with the captured user authentication data, and associates the captured user authentication data with the corresponding identification, *see, e.g.,* Application at 14, ¶ [0035], lines 1-4.

Claim 23 depends on claim 22 and adds the limitation that after populating the target datastore, further comprising prompting for and receiving from the user the user authentication data associated with the identification, *see, e.g.,* Application at 12, ¶ [0031], lines 1-6.

Claim 24 depends on claim 23 and adds the limitation of wherein the action of prompting for and receiving from the user the user authentication data associated with the identification comprises prompting for and receiving from the user the identification and the user authentication data associated with the identification, *see, e.g.,* Application at 12, [0031], lines 1-6.

Claim 25 depends on claim 22 and adds the limitation that migrating the source datastore to the target datastore comprises: reading unencrypted data from the source datastore, *see, e.g.,* Application at 9, ¶ [0021], lines 3-4, wherein the source datastore contains encrypted user authentication data and other unencrypted data, wherein the unencrypted data represents data for multiple users, *see, e.g.,* Application at 8, ¶ [0018], lines 5-7, each having an associated user identification and an associated user authentication data, *see, e.g.,* Application at 8, ¶ [0017], lines 1-5; converting the unencrypted data to be compatible with the target datastore, *see, e.g.,* Application at 12, ¶ [0029], lines 6-7; and populating the target datastore with the converted data, *see, e.g.,* Application at 12, ¶ [0029], liens 6-7.

Claim 26 depends on claim 22 and adds the limitation of authenticating the received identification using the target user authenticator upon determining that the target datastore includes user authentication data associated with the corresponding identification, *see, e.g.,* Application at 12, ¶ [0031], lines 1-6 and at 13, ¶ [0032], lines 1-7.

Claim 27 depends on claim 26 and adds the limitation that authenticating the received identification further comprises prompting for an receiving from the user the user authentication

data associated with the corresponding identification, *see, e.g.,* Application at 13, ¶ [0032], lines 1-7.

Claim 28 depends on claim 22 and adds the limitation that the target datastore is an LDAP compliant directory service, *see, e.g.,* Application at 7, ¶ [0016], lines 3-4.

Claim 29 depends on claim 22 and adds the limitation that the target datastore is a relational database, *see, e.g.,* Application at 4, ¶ [0006], lines 7-10.

Claim 30 depends on claim 22 and adds the limitation that the source datastore is a relational database, *see, e.g.,* Application at 4, ¶ [0006], line 6.

Claim 31 depends on claim 22 and adds the limitation that the user is a person, *see, e.g.,* Application at 4, ¶ [0006], lines 2-3 and at 8, ¶ [0018], line 7.

Claim 32 depends on claim 22 and adds the limitation that the user is a software object, *see, e.g.,* Application at 4, [0006], lines 2-3 and at 8, ¶ [0018], line 7.

Claim 33 depends on claim 22 and adds the limitation that the user authentication data is a password, *see, e.g.,* Application at 8, ¶ [0017], lines 1-5.

Claim 34 depends on claim 22 and adds the limitation that receiving user authentication from a user further comprises prompting for and receiving the identification and the user authentication data from the user after the initial submission of the identification from the user, *see, e.g.,* Application at 14, ¶ [0036], lines 1-6.

Claim 35 depends on claim 22 and adds the limitation that wherein the interceptor is a servlet, *see, e.g.,* Application at 9, ¶ [0022], line 2 to 10, ¶ [0022], line 6.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether claims 22 and 25-35 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Sampson, et al. (U.S. Pat. No. 6,429,624) (hereinafter "Sampson") in view of Blakley, III, et al. (U.S. Pat. No. 5,832,211) (hereinafter "Blakley").

B. Whether claims 23 and 24 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Sampson in view of Blakley and further in view of Mehring, et al. (U.S. Pat. No. 6,609,115) (hereinafter "Mehring").

VII. ARGUMENT

The Examiner has failed to establish a *prima facie* case of obviousness with respect to claims 22-35. All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). As noted by the United States Supreme Court in *Graham v. John Deere Co. of Kansas City*, an obviousness determination begins with a finding that **"the prior art as a whole in one form or another contains all" of the elements of the claimed invention**. *See Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 22 (U.S. 1966). Appellants respectfully submit that Sampson and Blakley, alone or in combination, do not contain all of the elements of claims 22-35. Further, Mehring does not cure the deficiencies of Sampson and Blakley.

Data migration as described throughout the pending disclosure, including paragraphs [0017]-[0026], and as used herein refers to the process by which data is moved from one proprietary system to another proprietary system. For example, the proprietary system by one vendor may store encrypted data using one proprietary encryption scheme while another vendor may use a different proprietary encryption scheme. The data migration process is not simply the copying of data from one source datastore to a target datastore, but rather includes the gathering of data from a source datastore in a first format, converting some or all of the data in the first format to a second format of the target datastore, and then saving the converted data in the second format on the target datastore. In addition, data migration requires that encrypted data elements of the source datastore, such as encrypted password information, which cannot be easily decrypted by the target datastore, be obtained from the user and added to the target datastore. For example, the target datastore may not have decryption keys for decrypting encrypted data elements of the source datastore. Obtaining encrypted data elements of the

source datastore from the user maintains consistency between the contents of the source datastore and the target datastore. This two step process overcomes many of the prior art limitations and does not require the decryption of the source datastore by the target datastore.

As disclosed in paragraph [0021], intercepting data may refer to the process by which a user sends password data intended for the source datastore and the target datastore intercepts the password data. Accordingly, user password information which is stored as encrypted user password information on the source datastore is obtained without decrypting the encrypted user password information on the source datastore. Unlike packet sniffing, interception includes obtaining the password data, verifying the password data, and storing password data in the target datastore. The user password data is intercepted directly from the user and can be verified using the source database. This verification occurs by having the target datastore send the password data to the source datastore, and the source datastore confirming that the password data is valid. Therefore, the target datastore performs the data interception by acting as an intermediary between a user and the source datastore. As described above, encrypted password data may be captured and provided to the target datastore using the interception process, thereby allowing for user passwords and other encrypted data to be migrated from a first proprietary system to a second proprietary system without having every user re-enter their security information.

Sampson relates to a system that controls access to information resources. A session manager enables a client to securely interact with a plurality of access servers and associated runtime elements using a plurality of sessions. The information resources are stored on protected servers. Access to the protected servers is controlled by one of the access servers. The session manager determines whether the client is involved in an authenticated session with any access server in the system. If so, the client is permitted to access the resources without logging

in to the specific access server that is associated with the protected server. In this way, the client can access multiple resources of multiple protected servers without logging in to each of the access servers that controls each of the protected servers. See, Sampson, Abstract. Sampson does not teach or suggest migrating the source datastore to a target datastore where the datastore comprises user identification data. In fact, Sampson is completely unconcerned with migrating data of any type at all. Sampson is simply concerned with providing a single sign-on for a user so that the user doesn't have to log in multiple times to access various information on the system. Such a system is unrelated to migrating authentication data from one datastore to another without requiring a user to re-enroll. Furthermore, such a system does not mitigate the inconvenience to users when migrating from one authentication system with encrypted passwords to a second authentication system since it does not provide a mechanism for populating the password data of the second authentication system without requiring the user to reregister.

Blakley relates to a network system server that provides password synchronization between a main datastore and a plurality of secondary datastores so that a user is able to maintain a single, unique password among the plurality of secondary datastores. See, Blakley, Abstract. Blakley uses a password synchronization server to store user names and plain-text passwords securely and to respond to requests from secondary datastores for their retrieval. See, Blakley, column 7, lines 32-35. Blakley does not teach or suggest migrating data from a source datastore to a target datastore, and does not address the problem of migrating from one vendor's proprietary encryption scheme to another vendor's product without having every user re-enter their security information and without decrypting the data stored in the first vendor's product.

Passwords are often stored encrypted by a hash algorithm. There is typically no decryption mechanism for such encrypted data. The only way to determine whether a password

is authentic is to compare a hash of the entered password to the stored encrypted password to determine if there is a match. Blakley requires a mechanism in which the data is decryptable. Accordingly, a system according to Blakely would not mitigate the inconvenience to users of having to register with a new authentication system because it does not provide a mechanism for populating the password data for the new authentication system when the encrypted password data in the old authentication system is non-decryptable.

Mehring relates to a method of allowing a remote system user to request multiple software applications using a single log-in. Although the remote user is only required to log-in once, the user information is submitted to the policy server every time the remote user logs-in to a different web server. See, Mehring, column 10, line 49 - column 11, line 24. Like Blakley, Mehring does not teach or suggest migrating data from a source datastore to a target datastore, and does not address the problem of migrating from one vendor's proprietary encryption scheme to another vendor's product without having every user re-enter their security information. Providing a single sign-on does not help when migrating encrypted password data from an old authentication system to a new authentication system since it does not address the problem of determining the passwords for users where the password data in the old system is not decryptable. It is respectfully submitted that Mehring does not cure the deficiencies of Blakley.

These distinctions, as well as others, will be discussed in greater detail in the analysis of the present claims that follows.

A. The rejection of claims 22 and 25-35 under 35 U.S.C. § 103(a) as rendered obvious by Sampson in view of Blakley was in error.

1. Sampson in view of Blakley does not teach or suggest capturing the password provided to the source user authenticator, monitoring the source user authenticator for an approval response and populating the

target datastore with the captured password upon receipt of an approval response.

Sampson in view of Blakley does not teach or suggest capturing the password provided to the source user authenticator, monitoring the source user authenticator for an approval response and populating the target datastore with the captured password upon receipt of an approval response.

Claim 22 of the pending application recites in part:

> using the source user authenticator to prompt for and receive the identification and a password from the user, the target user authenticator: monitors the source user authenticator for an approval response, and upon an approval response from the source user authenticator, captures the password provided to the source user authenticator by the user in response to prompting by the source authenticator, populates the target datastore with the captured password, and associates the captured password with the corresponding identification.

Thus, migrating password data from a source (or first) datastore to a target (or second) datastore according to the method of claim 22 may be accomplished without decrypting the password data stored in the source datastore and copying the decrypted password data to the target datastore. Rather, the interceptor captures the password provided by the user to the source user authenticator in response to prompting by the source user authenticator and waits to see whether the source user authenticator approves the user. If the source user authenticator approves the user's password, then the target user authenticator can populate its own datastore with the captured password knowing that the captured password is authentic. In some cases, the password data may be stored in the source datastore as, for example, a hash or other format that is not subject to being decrypted. However, such a method of storage is no impediment for migrating password data from the source datastore to the target datastore according to the method of claim 22 since no decryption is necessary. The only requirement is that the source

user authenticator is still functioning such that it is able to verify that the entered password is authentic.

The Final Office Action acknowledges that Sampson does not teach these elements of claim 22 recited above, but alleges that Blakley does disclose these elements in column 7, lines 17-35. *See*, Final Office Action, pp. 4-5. Appellants respectfully disagree. Blakley, column 7, lines 17-35 discloses synchronization of passwords between a DCE register and a foreign registry that has been configured into the network after the DCE registry has been populated with user definitions. Blakley, column 7, lines 17-35 further discloses that the synchronization provides a way for a foreign registry to receive the passwords for selected users without the need to wait for those points in time at which the passwords may be changed. In order to accomplish this, and in contrast to claim 22, Blakley requires that passwords stored in a first datastore be decrypted and then passed as plain text passwords to the second datastore. See, for example, Blakley, column 8, which states that "if the value is identified, the password synchronization server retrieves client W's password from the password repository and decrypts it. Next, in block **450**, the password synchronization server returns client W's password to foreign registry Z." This is a completely different method for transferring password data from one repository to another and is inapplicable to systems in which the password is stored as, for example, a hash, which is an item typically not capable of being decrypted. Specifically, Blakley does not teach capturing the password provided to the source user authenticator, monitoring the source user authenticator for an approval response and populating the target datastore with the captured password upon receipt of an approval response as claimed.

2. Neither Sampson nor Blakley teach or suggest migrating password data from a source datastore to a target datastore.

Claim 22 of the pending application recites "migrating the source datastore to the target datastore, wherein the source datastore comprises user identification data and user authentication data, wherein the source datastore is associated with a source user authenticator, wherein the target datastore is associated with a target user authenticator, and wherein the target user authenticator is in communication with the source user authenticator." The Final Office Action alleges that Sampson discloses this feature citing column 6, lines 2-5, column 7, lines 30-32, and column 17, lines 1-15. *See*, Final Office Action, p. 4. Appellants respectfully disagree with this assertion. In column 6, lines 2-5, Sampson discloses that each registry server may execute operations using multiple execution threads in which access of each thread to a registry repository is managed by an access control library. Sampson, in col. 7, lines 30-32, discloses that the registry server has an authentication server module that manages concurrent access of multiple users of browsers to the registry repository. Sampson further discloses, in column 17, lines 1-15, a computer system that includes a main memory for storing temporary variables or other intermediate information during executions of instructions to be executed by a processor, a static storage device for storing static information and instructions for processor, and a storage device for storing information and instructions. Nothing in these passages teach or suggest migrating the source datastore to the target datastore, wherein the source datastore comprises user identification data and user authentication data, wherein the source datastore is associated with a source user authenticator, wherein the target datastore is associated with a target user authenticator, and wherein the target user authenticator is in communication with the source user authenticator as claimed.

Rather, in contrast to the claims, Sampson is simply directed to a system that controls access to information resources. *See*, Sampson, Abstract. In particular, Sampson discloses a

session manager that determines whether the client is involved in an authenticated session with any access server in the system. *See, e.g.,* Sampson, Abstract. If so, the client is permitted to access the resources without logging in to the specific access server that is associated with the protected server. *See, e.g.,* Sampson, Abstract. Thus, Sampson merely teaches a single sign-on system such that a user only has to authenticate themselves once rather than multiple times even though the system may include multiple protected resources requiring authentication. Sampson does not disclose migrating identification and authentication (e.g., password) data from a source datastore to a target datastore. In fact, Sampson is completely unconcerned with migrating data of any type at all. Sampson is simply concerned with providing a single sign-on for a user so that the user does not have to log in multiple times in order to access various protected resources in the system. Such a system is unrelated to migrating authentication data from one datastore to another without requiring a user to re-enroll.

Blakley does not cure this deficiency in Sampson. Blakley provides password synchronization between a main data store and a plurality of secondary data stores. *See, e.g.,* Blakley, Abstract. This enables the user to maintain a single, unique password among the plurality of secondary datastores. Thus, Blakley uses a password synchronization server to store user names and plain-text passwords securely and to respond to requests from secondary datastores for their retrieval. The passwords are sent to the secondary datastores using encryption that is decipherable by the secondary datastores. However, notably absent from Blakley is any teaching or suggestion of migrating data from a source datastore to a target datastore. Note that data migration is not equivalent to data replication. Also, Blakley does not address the problem of migrating from one vendor's proprietary encryption scheme to another

vendor's product without having every user re-enter their security information and without decrypting the data stored in the first vendor's product.

3. Regarding claim 25, Sampson in view of Blakley does not teach or suggest converting unencrypted data from the source datastore to be compatible with the target datastore and populating the target datastore with the converted data.t

Claim 25 recites:

> reading unencrypted data from the source datastore, wherein the source datastore contains encrypted user authentication data and other unencrypted data, wherein the unencrypted data represents data for multiple users each having an associated user identification and an associated user authentication data;
> converting the unencrypted data to be compatible with the target datastore; and
> populating the target datastore with the converted data.

The Final Office Action states that Blakley discloses this feature relying on column 7, lines 30-60. *See*, Final Office Action, p. 6. Appellants respectfully disagree. Column 7, lines 30-60 of Blakely is directed to solving the problem of password synchronization between the DCE registry, which only stores encrypted passwords, and a foreign registry, configured into the network after the DCE registry has been populated with user definitions, where the synchronization occurs prior to an update of a given user's password. The details to the solution to this problem are described in Blakley, column 7, line 60 to column 8, line 52. Ultimately, Blakley discloses that a password is obtained from the password repository, decrypted, and the unencrypted password is returned to the requesting foreign registry. *See,* Blakely, col. 8, lines 48-52. Thus, Blakely does not disclose reading unencrypted data from the source and converting the unencrypted data to be compatible with the target datastore, but rather discloses decrypting encrypted data.

B. The rejection of claims 23 and 24 under 35 U.S.C. § 103(a) as being obvious over Sampson in view of Blakley and further in view of Mehring was in error.

1. Claims 23 and 24 depends directly or indirectly from claim 22 and are therefore allowable for at least the reasons established in sections VII.A.1. and VII.A.2.

Claim 23 depends directly from claim 22 and incorporates all of the limitations thereof. Claim 24 depends indirectly from claim 22 and incorporates all of the limitations thereof. Accordingly, for at least the reasons established in sections VII.A.1. and VII.A.2. above, Appellants respectfully submit that claims 23 and 24 are not taught or suggested by Sampson in view of Blakley. Appellants respectfully submit that Mehring does not cure the deficiencies in Sampson and Blakley. Therefore, Appellants respectfully submit that the rejection of claims 23 and 24 was in error and respectfully request that these claims be allowed.

VIII. CONCLUSION

In view of the above arguments the Appellants respectfully request that the Final Rejection of the claims be reversed and the case advanced to issue. Should the Examiner feel that a telephone interview would advance prosecution of the present application, the Appellants invite the Examiner to call the attorneys of record.

The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, of Sprint Communications Company L.P.

Respectfully submitted,
CONLEY ROSE, P.C.

Date: August 30, 2010

/Michael W. Piper/
Michael W. Piper
Reg. No. 39,800

ATTORNEY FOR APPELLANTS

CONLEY ROSE, P.C.
5601 Granite Parkway, Suite 750
Plano, Texas 75024
(972) 731-2288
(972) 731-2289 (facsimile)

IX. CLAIMS APPENDIX

1-15. (Canceled)

16. (Withdrawn) A method for populating password data in a target datastore of a target user authenticator after migration from a source user authenticator having a source datastore, while also responding to user requests for information, the method comprising:

migrating the source datastore to the target datastore, wherein the source datastore comprises user identification data and user authentication data, wherein the source datastore is associated with a source user authenticator, wherein the target datastore is associated with a target user authenticator, and wherein the target user authenticator is in communication with the source user authenticator;

intercepting, by an interceptor, a request to the source user authenticator from a user seeking access to information protected by the target user authenticator, wherein the interceptor prompts the user for an identification, receives the identification from the user, locates a corresponding identification in the target datastore, and searches the target datastore for a password associated with the identification; and

determining that there is no password in the target datastore associated with the identification:

forwarding the original intercepted request to the source user authenticator,

using the source user authenticator to prompt for and receive the identification and a password from the user,

capturing the password provided to the source user authenticator by the user in response to prompting by the source authenticator,

monitoring the source user authenticator for an approval response,

populating the target datastore with the captured password upon receipt by the target datastore of an approval response from the source user authenticator, and

associating the captured password with the corresponding identification.

17. (Withdrawn) The method of claim 16, wherein after populating the target datastore, further comprising prompting for and receiving from the user the password associated with the identification.

18. (Withdrawn) The method of claim 17, wherein the action of prompting for and receiving from the user the password associated with the identification comprises prompting for and receiving from the user the identification and the password associated with the identification.

19-21. (Canceled)

22. (Previously Presented) A method for populating password data to a target datastore of a target user authenticator after migration from a source user authenticator having a source datastore, while also responding to user requests for information, the method comprising:

migrating the source datastore to the target datastore, wherein the source datastore comprises user identification data and user authentication data, wherein the source datastore is associated with a source user authenticator, wherein the target datastore is associated with a target user authenticator, and wherein the target user authenticator is in communication with the source user authenticator;

intercepting, with an interceptor, a request to the source user authenticator from a user seeking access to information protected by the target user authenticator, wherein the interceptor prompts the user for an identification, receives the identification from the user, locates a corresponding identification in the target datastore, and searches the target datastore for a user authentication data associated with the corresponding identification;

forwarding the original intercepted request to the source user authenticator upon determining that the target datastore does not include a user authentication data associated with the corresponding identification;

using the source user authenticator to prompt for and receive the identification and user authentication data from the user, the target user authenticator: monitors the source user authenticator for an approval response, and upon an approval response from the source user authenticator, captures the user authentication data provided to the source user authenticator by the user, populates the target datastore with the captured user authentication data, and associates the captured user authentication data with the corresponding identification.

23. (Previously Presented) The method of claim 22, wherein after populating the target datastore, further comprising prompting for and receiving from the user the user authentication data associated with the identification.

24. (Previously Presented) The method of claim 23, wherein the action of prompting for and receiving from the user the user authentication data associated with the identification comprises prompting for and receiving from the user the identification and the user authentication data associated with the identification.

25. (Previously Presented) The method of claim 22, wherein migrating the source datastore to the target datastore comprises:

reading unencrypted data from the source datastore, wherein the source datastore contains encrypted user authentication data and other unencrypted data, wherein the unencrypted data represents data for multiple users each having an associated user identification and an associated user authentication data;

converting the unencrypted data to be compatible with the target datastore; and

populating the target datastore with the converted data.

26. (Previously Presented) The method of claim 22 further comprising:

authenticating the received identification using the target user authenticator upon determining that the target datastore includes user authentication data associated with the corresponding identification.

27. (Previously Presented) The method of claim 26, wherein authenticating the received identification further comprises prompting for an receiving from the user the user authentication data associated with the corresponding identification.

28. (Previously Presented) The method of claim 22, wherein the target datastore is an LDAP compliant directory service.

29. (Previously Presented) The method of claim 22, wherein the target datastore is a relational database.

30. (Previously Presented) The method of claim 22, wherein the source datastore is a relational database.

31. (Previously Presented) The method of claim 22, wherein the user is a person.

32. (Previously Presented) The method of claim 22, wherein the user is a software object.

33. (Previously Presented) The method of claim 22, wherein the user authentication data is a password.

34. (Previously Presented) The method of claim 22, wherein receiving user authentication form a user further comprises:

prompting for and receiving the identification and the user authentication data from the user after the initial submission of the identification from the user.

35. (Previously Presented) The method of claim 22, wherein the interceptor is a servlet.

X. EVIDENCE APPENDIX

None.

XI. RELATED PROCEEDINGS APPENDIX

None.